Master 1 Informatique - Réseaux et Communications - Projet
La webcam du Grand Lyon - version simplifiée

Ce projet vous permet de mettre en œuvre des outils vus en cours : communications entre processus sur internet, threads, moyens de synchronisation, etc. Il est inspiré d'une application réelle, en voici un aperçu.

1 Description générale

Le site internet du Grand Lyon offre aux internautes la possibilité de piloter en temps réel une caméra installée au sommet de la tour Part-Dieu, dans le quartier d'affaires de la Part Dieu à Lyon. Les internautes peuvent ainsi avoir une vue sur la ville de Lyon.

Les internautes connectés au serveur reçoivent les images en temps réel et pour une durée maximum de 10 minutes. Pendant cette durée, ils peuvent aussi demander le pilotage de la webcam pour une durée maximum de 60 secondes. Un seul internaute à la fois peut avoir la main sur la webcam pour le pilotage et peut perdre la main au bout des 60 secondes s'il n'y a pas d'autres internautes en attente. Dès qu'un internaute obtient la main sur la webcam, il peut envoyer ses commandes de pilotage.

2 Adaptation pour le projet

Dans ce projet il ne s'agit pas de réaliser des transferts de flux vidéo ni de mettre l'accent sur la gestion du temps. Pour simplifier, nous supposons les deux points suivants :

1. la donnée transmise est une grille (au départ vide) de taille limitée. Dans cette grille, un internaute peut déplacer un curseur à partir de son emplacement courant. Chaque mouvement du curseur est matérialisé par l'apparition d'un symbole dans une case voisine respectant la direction choisie par l'internaute. Les directions possibles sont vers la droite, gauche, haut, bas ou en diagonale. Afin d'éviter les débordements, chaque mouvement à l'extérieur de la grille est considéré sans effet. Nous supposons aussi que la grille est réinitialisée après un certain nombre de mouvements effectués par les internautes.
2. que le temps de connexion des clients, y compris pour le pilotage est illimité, toutefois, qu'un client ne reste pas indéfiniment connecté. **Remarque :** voir la question bonus pour la gestion du temps.

Le programme comprend un serveur et deux types de clients : ceux qui pilotent et ceux qui visualisent uniquement la grille. Le serveur doit être capable de diffuser l'état de la grille, en temps réel, à tous les clients connectés et de gérer, une par une, les demandes de pilotage.

3 Guide pour la réalisation du projet

Les étapes suivantes vous guident dans la réalisation du projet. Il est important de les suivre pour faciliter la progression dans la réalisation des différentes fonctionnalités demandées.

3.1 Pour commencer

Définir l'architecture de votre application (décomposition en processus et en threads, avec le rôle de chacun) et le choix des outils à utiliser. Cette partie devra paraitre dans votre rapport avec justification de vos choix. Ne commencez pas la programmation tant que cette étape n'est pas réalisée. La lecture intégrale et attentive de ce sujet vous permettra de faire vos choix. Vous devez notamment être attentifs aux traitements des connexions et différentes requêtes, aux accès concurrents, etc.

Cette étape vous permet aussi de répartir les tâches à réaliser.

3.2 Protocole d'échanges entre les clients et le serveur

Pour que les clients et le serveur puissent "dialoguer" correctement, il faut organiser tous les échanges d'informations. Pour cela, il faut définir un protocole qui détermine le contenu des messages échangés entre un client et le serveur.

Définir un protocole pour la connexion/déconnexion des clients au serveur, les requêtes des clients et les réponses du serveur (y compris comment la grille est transmise).

Le protocole d'échange devra paraitre dans votre rapport de projet avec justification de vos choix. Enfin, comme protocole de communication, vous utiliserez TCP.

3.3 Visualisation de la grille par les clients

Cette première étape de programmation se concentre sur la diffusion de la grille sans pilotage par les clients. Nous supposons qu'une fonction modifiant aléatoirement la grille s'exécute sur le serveur. Les clients se contentent donc de visualiser l'évolution de cette grille. Vous pouvez aussi commencer par une configuration avec un seul client avant de passer à une version concurrente du serveur.

3.4 Pilotage

Enrichir votre programme de manière à ce que le serveur puisse gérer les demandes de pilotage effectuées par des clients ainsi que le pilotage. Un client qui pilote ou qui est en attente d'avoir la main pour le pilotage, doit aussi pouvoir visualiser l'évolution de la grille.

4 Précisions générales

- Dans l'ensemble des étapes de conception et de programmation, vous devez faire attention aux éventuels accès concurrents à des données et à la synchronisation.
- Vous pouvez vous contenter d'interfaces textes pour les clients. L'implémentation d'une interface graphique sera prise en compte dans la notation du projet, uniquement si l'ensemble des fonctionnalités demandées et le rapport sont correctement réalisés.
- Il est nécessaire d'implémenter un serveur concurrent en utilisant des threads. A vous de voir comment les utiliser à bon escient.

5 Améliorations (optionnelles)

Pour améliorer votre note, il est bien entendu possible de suggérer et d'implémenter des fonctionnalités supplémentaires. Toutefois, les améliorations que vous réaliserez ne seront pas prises en compte en cas de problèmes dans la réalisation des parties précédentes (erreurs à la compilation/exécution, cas non traités, code illisible, etc.).

Parmi les améliorations possibles, il y a la réalisation d'une interface graphique et/ou la gestion du temps de connexion (pour pilotage et/ou visualisation) des clients par le serveur.

6 Document à rendre

L'ensemble des documents à rendre sera sous forme d'une archive *nomEtudiant1-nomEtudiant2*.tar.gz, avec :

- un rapport (au format PDF) avec : mode d'emploi, description de l'architecture de votre application (justifiée), protocoles de d'échange (justifiés), schémas algorithmiques de vos programmes clients et serveur, difficultés rencontrées et solutions apportées (les problèmes du genre difficulté de gestion de temps de travail et de programme chargé, etc. n'ont pas à y paraitre!!), etc. Le rapport ne doit pas excéder 6 pages.
- l'ensemble des sources accompagné d'un Makefile.

L'archive est à déposer en utilisant l'espace pédagogique. Si vous n'êtes pas encore inscrit dans le module à partir de cet espace, faites le.

Attention : aucun changement dans le code ne doit être fait par le correcteur avant de pouvoir exécuter vos programmes. Pensez au passage de paramètres!

7 Déroulement, dates et soutenance

Le travail est à effectuer par binômes! Les binômes devront être annoncés en début de la première séance du projet (semaine du 28/11).

Votre présence en séance de TP pour la réalisation du projet est obligatoire pour votre suivi. Durant ces séances (et non en dehors), vous pouvez poser des questions à votre encadrant.

Date limite de dépôt des documents, le 23/12/2010 à 20h00. Après ce délai, AUCUN dépôt ou envoi par email ne sera accepté et la note sera 0/20.

Une soutenance orale se fera durant la semaine du 03/01/11 le mardi et vendredi pendant les heures de TD/TP. Vous aurez 10 minutes pour faire une démonstration de votre projet en utilisant deux à trois machines des salles de la FdS, et pour répondre à des questions.